WHAT IS CLAIMED IS:

1. A method for making a colorful three dimensional model comprising steps of:

inputting three dimensional original measured data;

reconstructing mesh models with regular data;

abstracting color information;

harmonizing color of texture images; ; and

pixel blending to overlapped texture images between the mesh models.

2. The method as claimed in claim 1, wherein the mesh model reconstructing step comprises:

selecting a generic model according to the original measured data;

adjusting dimension and spatial position of the generic model to overlap with the original measured data; and

mapping data of the generic model with the original measured data to deform the generic model data to be close to the original measured data.

3. The method as claimed in claim 1, wherein the color abstracting step is to establish texture-mapping relationship between two dimensional image of the original measure data and the generic model, which comprises:

seeking mapping points of mesh points of the generic model on the original measured data and triangles having the mapping points;

calculating corresponding texture coordinates of the mapping points; and

checking continuity of the triangles on the texture images.

4. The method as claimed in claim 1, wherein the color harmonizing step comprises:

rearranging sequence of measured data according to the overlapped relationship

and the magnitude of the overlapping area to be $M' = \{M'_1, M'_2, \ldots, M'_n\}$, wherein M'_n represents data consisting of n three dimensional mesh models M';

calculating color adjustment A_i (i=1,2,3...n) of the texture image of each original measured data; and

adjusting color average of the overlapped area.

5. The method as claimed in claim 2, wherein the color harmonizing step comprises:

rearranging sequence of measured data according to the overlapped relationship and the magnitude of the overlapping areat to be $M' = \{M'_1, M'_2, \ldots, M'_n\}$, wherein M'_n represents data consisting of n three dimensional mesh models M';

calculating color adjustment A_i (i=1,2,3...n) of the texture image of each original measured data; and

adjusting color average of the overlapped area.

6. The method as claimed in claim 3, wherein the color harmonizing step comprises:

rearranging sequence of measured data according to the overlapped relationship and the magnitude of the overlapping area to be $M' = \{M'_1, M'_2, \ldots, M'_n\}$, wherein M'_n represents data consisting of n three dimensional mesh models M';

calculating color adjustment A_i (i=1,2,3...n) of the texture image of each original measured data; and

adjusting color average of the overlapped area.

7. The method as claimed in claim 4, wherein the color harmonizing step comprises:

rearranging sequence of measured data according to the overlapped relationship

and the magnitude of the overlapping area to be $M' = \{M'_1, M'_2, \ldots, M'_n\}$, wherein M'_n represents data consisting of n three dimensional mesh models M';

calculating color adjustment A_i (i=1,2,3...n) of the texture image of each original measured data; and

adjusting color average of the overlapped area.

8. The method as claimed in claim 4, wherein $A_i = (A_{i,1} \times W_{i,1} + \ldots + A_i, A_{i-1} \times W_i, W_{i-1}) / (W_{i,1} + \ldots + W_{i,i-1})$

where W_i is mesh influenced weight value.

9. The method as claimed in claim 5, wherein $A_i = (A_{i,1} \times W_{i,1} + \ldots + A_i, A_{i-1} \times W_i, W_{i-1}) / (W_{i,1} + \ldots + W_{i,i-1})$

where W_i is mesh influenced weight value.

10. The method as claimed in claim 6, wherein $A_i = (A_{i,1} \times W_{i,1} + \ldots + A_i, A_{i-1} \times W_i, W_{i-1}) / (W_{i,1} + \ldots + W_{i,i-1})$

where W_i is mesh influenced weight value.

11. The method as claimed in claim 7, wherein $A_i = (A_{i,1} \times W_{i,1} + \ldots + A_i, A_{i-1} \times W_i, W_{i-1}) / (W_{i,1} + \ldots + W_{i,i-1})$

where W_i is mesh influenced weight value.

12. The method as claimed in claim 1, wherein the pixel blending step to the overlapped texture image comprises:

seeking the overlapped images covered by each triangle within overlapped areas;

calculating distances of vertices of each of the triangles within the overlapped areas to nearest edges of corresponding mesh; and

calculating pixel weight average to mapping area corresponding to each

triangle.

13. The method as claimed in claim 2, wherein the pixel blending step to the overlapped texture image comprises:

seeking the overlapped images covered by each triangle within overlapped areas;

calculating distances of vertices of each of the triangles within the overlapped areas to nearest edges of corresponding mesh; and

calculating pixel weight average to mapping area corresponding to each triangle.

14. The method as claimed in claim 3, wherein the pixel blending step to the overlapped texture image comprises:

seeking the overlapped images covered by each triangle within overlapped areas;

calculating distances of distal points of each of the triangles within the overlapped areas to nearest edges of corresponding mesh; and

calculating pixel weight average to mapping area corresponding to each triangle.

15. The method as claimed in claim 4, wherein the pixel blending step to the overlapped texture image comprises:

seeking the overlapped images covered by each triangle within overlapped areas;

calculating distances of vertices of each of the triangles within the overlapped areas to nearest edges of corresponding mesh; and

calculating pixel weight average to mapping area corresponding to each

triangle.

16. The method as claimed in claim 8, wherein the pixel blending step to the overlapped texture image comprises:

seeking the overlapped images covered by each triangle within overlapped areas;

calculating distances of vertices of each of the triangles within the overlapped areas to nearest edges of corresponding mesh; and

calculating pixel weight average to mapping area corresponding to each triangle.

17. The method as claimed in claim 11, wherein the pixel blending step to the overlapped texture image comprises:

seeking the overlapped images covered by each triangle within overlapped areas;

calculating distances of vertices of each of the triangles within the overlapped areas to nearest edges of corresponding mesh; and

calculating pixel weight average to mapping area corresponding to each triangle.